

(1) Japanese Patent Application Laid-Open No. JP63-318632(1988)

“High-Speed Square-Root Extraction Operation Unit ”

The following is an extract relevant to the present application.

5

A high-speed square-root extraction operation device including an adder performing a binary addition of plural stages, a multiplexer of plural stages selecting an input signal of said adder, and a means for controlling said multiplexer based on a result of addition at the previous stage.

10

⑩ 日本国特許庁(J P)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭63-318632

⑬ Int. Cl.<sup>4</sup>

G 06 F 7/552

識別記号

庁内整理番号

B-7056-5B

⑭ 公開 昭和63年(1988)12月27日

審査請求 未請求 発明の数 1 (全5頁)

⑮ 発明の名称 高速開平演算装置

⑯ 特 願 昭62-155828

⑰ 出 願 昭62(1987)6月23日

⑱ 発 明 者	豊 蔵	真 木	大阪府門真市大字門真1006番地	松下電器産業株式会社内
⑱ 発 明 者	山 田	晴 保	大阪府門真市大字門真1006番地	松下電器産業株式会社内
⑱ 発 明 者	森	俊 樹	大阪府門真市大字門真1006番地	松下電器産業株式会社内
⑲ 出 願 人	松下電器産業株式会社			大阪府門真市大字門真1006番地
⑳ 代 理 人	弁理士 中尾 敏男			外1名

明 細 書

1、発明の名称

高速開平演算装置

2、特許請求の範囲

複数段の2進数加算を行なう加算器を有する開平演算装置であって、前記加算器の入力信号を選択する複数段のマルチプレクサを具備し、前段の加算結果により前記マルチプレクサを制御する手段を有することを特徴とする高速開平演算装置。

3、発明の詳細な説明

産業上の利用分野

本発明はデジタル信号処理プロセッサにおいて、平方根の演算を行なう開平演算装置に関する。

従来の技術

第3図に一般的非回復型の平方根の導出アルゴリズムを示す。(コンピュータの高速演算方式; HWANG, E 堀部ひさし訳 近代科学社 1980, 09参照)

以下、第3図に基づき、このアルゴリズムを説明する。

2つの正の2進数A、Qを次のように表す。

$$Q = \sqrt{A} = 0.q_1q_2 \dots q_n$$

$$A = Q^2 = 0.a_1a_2 \dots a_{2n-1}$$

まず、 $D_0 = 0.01$ とおき $a_1a_2$ から減ずる。残り $R_1$ が正の時 $q_1 = 1$ と置き、次の対 $a_3a_4$ を $R_1$ に追加し、それから $D_1 = 0.q_101$ を減ずる。 $R_1$ が負ならば $q_1 = 0$ と置き、 $a_3a_4$ を $R_1$ に追加し、 $D_1 = 0.q_111$ をそれに加える。この結果を $R_2$ と置き、次の処理のための判断の対象となる。同様に処理が進められるが、一般に $k$ 番目のステップでは、 $k$ 番目の平方根ビット $q_k$ が決められたのち、次の操作を行なうことになる。

$$R_k \leftarrow R_k \cdot a_{2k+1}a_{2k+2} - q_1q_2 \dots q_k 01 \quad q_k = 1 \text{ のとき}$$

$$R_k \leftarrow R_k \cdot a_{2k+1}a_{2k+2} + q_1q_2 \dots q_k 11 \quad q_k = 0 \text{ のとき}$$

ここで追加操作“ $\cdot$ ”は1つ前の残余を2ビット式ヘシフトし、右端へ新しい対を補充することによって表現される。

このアルゴリズムを実現するためには、 $q_k$ が0か1かに応じて減算か加算かを選択することが可能な制御可能加算減算 (Controlled Add -

Subtractor: OAS)セルを用いる。OASセルは、フルアダーとXORゲートから構成されている。8ビット2進数の平方根を4ビット2進数として計算するセル構成において、OASセルを20個必要とする。フルアダーを8個のゲートで構成すると、その他のインバータ3個と合わせて123個のゲートが必要となる。

発明が解決しようとする問題点

しかしながら上記のような構成では、前段の計算が終了するまで次段の計算は始まらないため、計算時間が遅く、また、OASセルの構成によるため、多くの素子数が必要となっていた。

本発明はかかる点に鑑み、少ない素子数で、高速に平方根演算を行なう開平演算装置を提供することを目的とする。

問題点を解決するための手段

本発明は、複数段の2進数加算器とその入力信号を前段の加算結果により選択するマルチプレクサを備えた開平演算装置である。

作用

1) 残余  $R_k$  が負のとき

$q_k = 0$  と置き、 $R_{k-1}$  に被開平方数の次の2ビット分の  $a_{2k+1}a_{2k+2}$  を追加し、これを  $R_k$  に置く。(ステップ5)

3)  $D_k = q_1 q_2 \dots q_k 01$  と置き、 $R_k - D_k$  を計算し、 $R_{k+1}$  に置く。そして  $k$  に1を加える。(ステップ6, ステップ7)

4) 繰返し回数  $k$  を調べる。(ステップ8)

1)  $k$  が  $n$  を超えなければ2)へ戻り、実行を進める。

1)  $k$  が  $n$  を超えれば、平方根  $Q$  を

$$Q = q_1 q_2 \dots q_n$$

とし、処理を終える。(ステップ9) このとき残余は、 $R_{n-1}$  である。ここで  $n$  は必要とする平方根の桁数である。

以上で述べたアルゴリズムを実際のハードで構成するための前段階として実行手順を第4図の並算例に従って説明する。第4図では2進8桁の10000000について開平演算を行ない、4桁の平方根と8桁の残余を求めている。まず

本発明は前記した構成により、それぞれの段の加算を途中まで実行しておくことが可能となり、また、各段の演算は、加算(減算を2の補数の加算で実現している)のみでよく、演算を加算か減算かを制御する必要がないため、素子数の削減が可能となる。

実施例

以下本発明の一実施例を図面に基づいて説明する。まず、本発明の開平演算装置の基礎となる開平演算方式のアルゴリズムを第2図に示す。以下第2図に従って本演算方式のアルゴリズムを述べる。

1) 初期設定として  $k=1$ ,  $R_0 = a_1 a_2$ ,  $D_0 = 01$  と置く。 $R_0 - D_0 = a_1 a_2 - 01$  を計算し、 $R_1$  とする。(ステップ1, ステップ2)

2) 残余  $R_k$  の値を調べる。(ステップ3)

1) 残余  $R_k$  が0または正のとき

$q_k = 1$  と置き、 $R_k$  に被開平方数の次の2ビット分の  $a_{2k+1}a_{2k+2}$  を追加し、これを  $R_k$  に置く。(ステップ4)

$a_1 a_2$  に11を加える。これは01を減ずることの代わりに2の補数を利用している。この加算の結果キャリーがあれば、 $q_1 = 1$ 、なければ  $q_0 = 0$  とする。今、キャリーがあり  $q_1 = 1$  とする。次に、加算の結果01に  $a_3 a_4 = 00$  を追加した0100に  $0 q_1 01$  の補数  $1 \bar{q}_1 11 = 1011$  を加える。この和は1111でキャリーはない。このとき  $q_2 = 0$  とする。さらに次のステップでは、 $q_2 = 0$  であるため一段前の加算結果100を対象とし、 $a_5 a_6 = 00$  を追加した10000に前と同様に10111を加えた結果00111とキャリーを得、 $q_3 = 1$  とする。最後のステップでは0111に  $a_7 a_8 = 00$  を追加した011100に101011を加えた結果000111とキャリーを得、残余00111及び平方根1011を得ることになり演算を終える。即ち10000000(2)=128(10)の平方根は1011(2)=11(10)であり残余は111(2)=7(10)となる。

次にこれまで述べてきた開平演算の実行手順を実現するための本発明の高速開平演算装置の一実

施例について以下で説明する。第1図が2進8桁の被開平数 $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ を2進4桁の平方根 $q_1, q_2, q_3, q_4$ に計算する例を示すものである。第1図において、101~110はフルアダー、201~214はマルチプレクサ、301~310はインバータ、401~403はORゲートであり、 $a_1 \sim a_8$ は8桁の被開平数入力、 $q_1 \sim q_4$ は4桁の平方根出力、 $r_1 \sim r_5$ は5桁の残余出力である。

実行手順に従い、加算はフルアダーで行ない、その結果、最上位の桁上げの値により加算結果と前段の結果のどちらかを用いるかの選択はマルチプレクサを用いている。

このヘッド構成において $a_1, a_2 + 11 = k_1, k_2, k_3$ の計算で $k_1, k_2$ は $a_1 + a_2 + 1$ に一致することを利用してフルアダー101の入力に“1”と $a_1$ と $a_2$ を入力して結果を得る。また $k_3$ は $\bar{a}_2$ であるのでインバータ301で $a_2$ を反転している。 $k_1$ は出力 $q_1$ として用いられ、 $k_2, k_3$ か $a_1, a_2$ かの選択はマルチプレクサ201及び202を用いて制御信号

$a_1 = "1"$ のとき $a_2, a_3, a_4, a_5$ は残余となるが、上記 $n=2$ の場合に相当し、 $2^5=1000$ を超えない。即ち $a_5=0$ となる。

次段では $a_1$ により $a_2, a_3, a_4$ と $a_3, a_4, a_5$ のどちらかが選択され計算の対象となり、以下同様に第4図に示す計算を行なうことにより、平方根 $q_1 \sim q_4$ 及び残余 $r_1 \sim r_5$ を求めることができる。

また、第1図ではフルアダー10個及びマルチプレクサ14個、ORゲート3個、インバータ10個で開平装置を構成している。フルアダーを6個のゲート及びマルチプレクサを4個のゲートで構成すると総ゲート数は129個となる。CASを使用した場合の143個に比べ14個のゲート数が削減される。さらに必ず1を入力するフルアダー等でゲートの節約及び残余を必要としない場合、最下段のマルチプレクサの節約が可能である。演算速度はフルアダーの通過回数を考慮すると本演算方式では最長ルートで7回であり、CASを使用する方法がシリアル的に20回であり、2倍以上の速度改善がなされる。2n桁の2進数から

$k_1$ により行なわれる。 $k_1$ が“1”のとき $k_2, k_3$ が選ばれ、 $k_1$ が“0”のとき $a_1, a_2$ が選ばれ $k_1, k_2$ として次段で用いられる。次段では、 $a_1, a_2, a_3, a_4 + 1, \bar{a}_1, 11 = a_1, a_2, a_3, a_4, a_5$ の計算が行なわれる。 $a_3, a_4 + 11$ の計算は初段と同様に行なわれ、その桁上げ $a_1$ と $a_2$ と $\bar{a}_1$ との加算がフルアダー102で行なわれる。この加算結果の桁上げ $a_2$ と $a_1$ と“1”との3入力の加算が必要となるが、1つの入力が“1”であるのでこの加算結果の桁上げ $a_1$ は $a_2$ と $a_1$ との論理和で計算されるので $a_2$ と $a_1$ を入力とするORゲート401で作られる。 $a_2$ は $q_2$ が“1”のときのみ次段で選択され用いられるべきであるが、 $a_1 (=q_2)$ が“1”のときは必ず“0”となることが以下で示すようにわかるので必要でない。

2n桁の2進数の平方根はn桁で表され、その残余が最も大きくなるのは、2n桁の2進数の最大値 $2^{2n}-1$ の平方根 $2^n-1$ が算出される場合であり、その残余は、 $2^{2n}-1-(2^n-1)^2=2^{n+1}-2$ となり、 $2^{n+1}$ を超えない。従って

n桁の2進数として開平演算をする場合、ゲート数は、CASを使用したものが $7n^2+8n-1$ 個に対し本発明では $\frac{11}{2}n^2+\frac{21}{2}n-1$ 個、フルアダー通過回数は、CASを使用したものが $n^2+n$ 回であるのに対し、本発明では $\frac{1}{2}n^2-\frac{1}{2}n+1$ 回となり、nが大きくなると素子の削減割合は $\frac{1}{14}=21.4\%$ 、フルアダー通過回数は $\frac{1}{2}$ 回に近くなる。

#### 発明の効果

以上述べたように、本発明によれば、開平演算を並列演算方式で実現する場合、素子数の削減と演算速度は2倍以上にすることができ、この実用的効果は大きい。

#### 4. 図面の簡単な説明

第1図は本発明の開平演算方式の一実施例のブロック図、第2図は本発明の非回復型の平方根の導出アルゴリズムのフローチャート図、第3図は従来の一般的な非回復型の平方根の導出アルゴリズムのフローチャート図、第4図は本発明における開平演算方式の実行手順の演算形式による説明図である。

101~110……フルダ、201~214

……マルチプレクサ。

代理人の氏名 弁理士 中 尾 敏 男 ほか1名

101-110 -- 7279-(F.A.)

201-214 -- マルチプレクサ(MPX)

८१

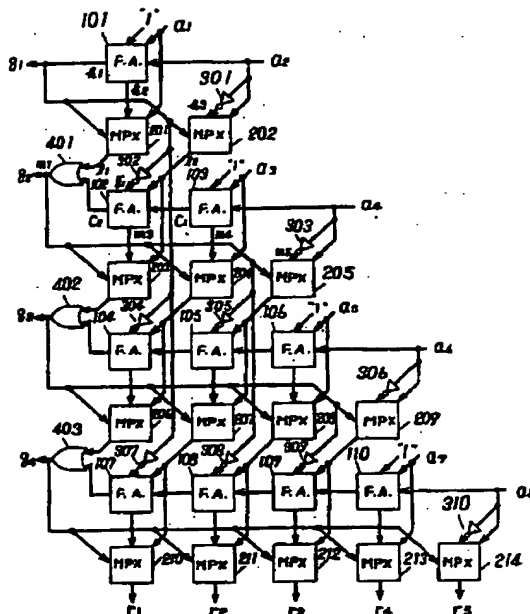
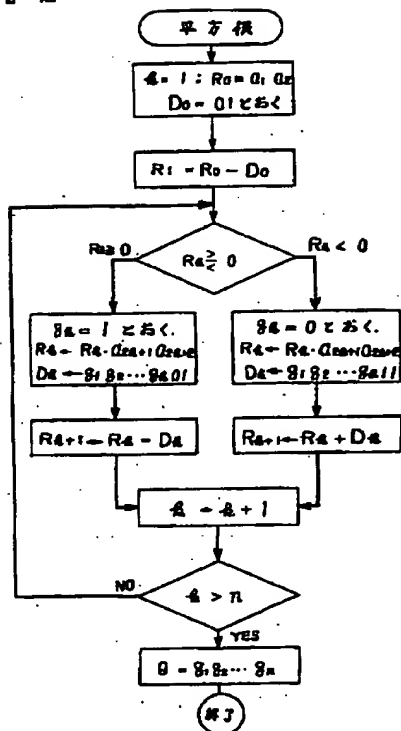
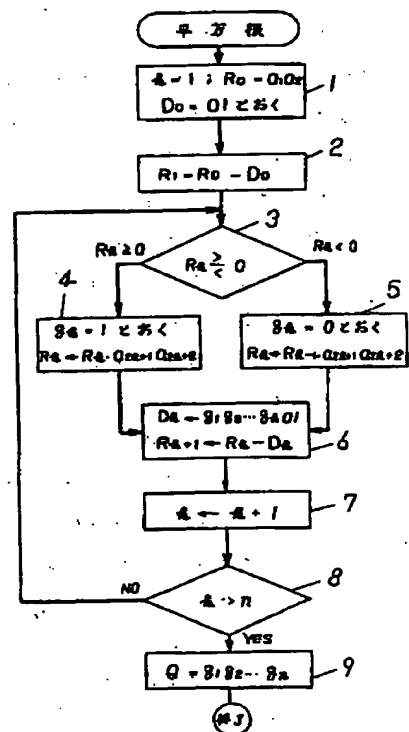


圖 2 示



第 3 圖



第 4 図

	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>	Q <sub>8</sub>
$\sqrt{\quad}$	1	0	0	0	0	0	0	0
+) $\overline{\quad}$	1	1						
$g_1 = 1$	0	1	0	0				
+) $\overline{\quad}$	1	0	1	1				
$g_2 = 0$	1	1	1	1				
		1	0	0	0			
+) $\overline{\quad}$	1	0	1	1	1			
$g_3 = 1$	0	0	1	1	1	0	0	
		1	0	1	0	1	1	
$g_4 = 1$	0	0	0	1	1	1	1	